REMARKS

Claims 1-23 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(b) Rejection:

The Examiner rejected claims 1, 7, 8, 13-15, 17, 18 and 23 under 35 U.S.C. § 102(b) as being anticipated by Mendelson et al. (U.S. Publication 2002/0095553) (hereinafter "Mendelson"). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Mendelson fails to teach or suggest *the prefetch unit is configured to fetch instruction code from a system memory for storage within the instruction cache, and wherein the prefetch unit is further configured to fetch a line of instructions into the instruction cache in response to a trace being evicted from the trace cache*. In the previous Office Action, the Examiner asserted that the L2 cache of Mendelson stores instructions and is, thus, analogous to the instruction cache of Applicants' claim 1. Applicants argued that L2 cache 340 of Mendelson is a second-level trace cache, that may be added to trace cache subsystem 120, and that is distinguished from a traditional L2 cache in that the L2 trace-cache 340 contains traces whereas a traditional L2 cache contains data or instructions that have not been built into a trace. (See, e.g., paragraphs [0036] – [0037]). Thus, Applicants argued that L2 trace-cache 340 is not an instruction cache, as would be understood by one of ordinary skill in the art, but a trace cache. The system of Mendelson does, however, include an instruction cache, specifically an L1 instruction cache. This instruction cache is illustrated as cache memory 140 of FIG. 1C (see, e.g., paragraph [0020].)

In the Response to Arguments section of the Office Action mailed September 7, 2006, the Examiner again asserts, "the L2 cache 340 does include instructions and the anticipation of Applicants' instruction cache, as claimed, is perfectly reasonable."

Applicants again assert that the term "instruction cache" is well known to those of ordinary skill in the art and has a specific meaning defined by its usage. It does not refer to any cache that may include instructions, as the Examiner suggests. Applicants refer the Examiner to Mendelson, which makes a clear distinction between an instruction cache (such as cache memory 140) and a trace cache (such as FTC 320, MTC 330, or L2 cache 340). For example, FIG. 1C illustrates cache memory 140 (which is one component of CPU front-end 160) separate from trace cache memory subsystem 120 (which includes FTC 320, MTC 330, and L2 cache 340), and paragraphs [0019] - [0021], reproduced below, describe these two distinct types of caches.

> [0019] The trace builder 110 exams dynamic instruction stream 106, selects and collates common instruction sequences. The sequences are fetched from the cache memory 140 and are built into traces 101.sub.1, to 101.sub.N ("N" being a positive whole number). The traces 101.sub.1 to 101.sub.N are built by an iterative process. In the iterative process, instructions are fetched from the CPU front-end 160, decoded and executed by the execution unit 130, and added to a trace cache subsystem 120 (see FIG. 1C) by the trace builder 110. The building of a trace depends mainly on the termination criterion, the test that determines where the trace ends and a new trace begins. The criterion may be a composite criterion with static/fixed or dynamic/adaptive conditions and configurable parameters FIG. 1C is a diagram illustrating the CPU front-end 160 in FIG. 1B according to one embodiment of the present invention. The CPU front-end 160 includes a cache memory 140, a trace builder 110, and a trace cache subsystem 120.
>
> [0020] In one embodiment, the cache memory 140 is an instruction cache that is built into the architecture microprocessors. The cache memory 140 may be an internal cache from a plurality of internal caches of the architecture microprocessors. It is noted that this type of cache is sometimes called Level 1 (L1) cache.
>
> [0021] The trace cache subsystem 120 provides the system 100 with an improved processor performance and lower power consumption. Higher performance is due to the higher instruction delivery and fewer pipeline stages. Lower power is due to the saving of repeated operations. The trace cache subsystem 120 takes traces 101.sub.1 to 101.sub.N and stores them in one of several trace caches located in the trace cache subsystem 120.

It is clear that the system of Mendelson considers the instruction cache 140 and the trace caches (in trace cache subsystem 120) to be distinct types of caches, which store different types of information and which are used for different purposes. For example, paragraph [0019] describes that instruction sequences are fetched from instruction cache

140 and are built into traces 101.sub.1, to 101.sub.N. Paragraph [0021] then describes that these traces (which, as described above, are built from, but are not the same as, the instruction sequences fetched from instruction cache 140) are stored in one of several trace caches located in trace cache subsystem 120. **Applicants assert that the Examiner's interpretation that L2 cache 340 of Mendelson anticipates the instruction cache of Applicants' claims is clearly not reasonable in light of the fact that the Examiner's cited reference itself makes a clear distinction between these two types of caches.**

In the previous Office Action, the Examiner further submitted that the Cache Manager 310 of Mendelson is analogous to Applicants' prefetch unit. Applicants argued that Cache Manager 310 is clearly not the same as the prefetch unit of Applicants' claim 1. First, the Cache manager is clearly not configured to fetch instruction code from a system memory for storage within the instruction cache, as recited in claim 1. Furthermore, a prefetch unit, such as that of Applicants' claim 1, is well known in the microprocessor art and one of ordinary skill in the art at the time the invention was made would not consider Cache Manager 310 of Mendelson to be a prefetch unit.

The Examiner submitted that L2 cache 340, Cache Manager 310, and paragraph [0039] of Mendelson disclose *wherein the prefetch unit is configured to fetch a line of instructions into the instruction cache in response to a trace being evicted from the trace cache.* The Examiner asserted, "the management logic fetches instruction traces to the L2 cache in response to their being evicted from a higher-level trace cache 320 or 330." Applicants argued that while this paragraph does describe that traces evicted from FTC 320 or MTC 330 are transferred for storage in the L2 trace cache (under control of Cache Manager 310), this has nothing to do with a prefetch unit fetching a line of instructions into the instruction cache in response to a trace being evicted from the trace cache. First, as discussed above, Cache Manager 310 is clearly not a prefetch unit, according to the limitations recited in Applicants' claim 1. Second, Cache Manager 310 does not move (or fetch) anything into the instruction cache (cache 140 of Mendelson), but instead transfers a trace entry into another trace cache (Mendelson's L2 trace cache). In addition, Cache

Manager 310 does not fetch a line of instructions (such as would be stored in system memory), in response to a trace being evicted, but instead transfers the evicted trace entry itself from one trace cache to another.

Applicants noted that a mechanism, similar to Mendelson's, for transferring an evicted trace to another level of trace cache is described in the specification of the present invention (see, e.g., paragraphs [0046]-[0047]). However, this mechanism is clearly not the subject matter of Applicants' claim 1, in which the prefetch unit, (i.e., one that is configured to fetch instruction code from a system memory for storage within the instruction cache), is configured to fetch a line of instructions into the instruction cache in response to a trace being evicted from the trace cache.

Applicants note that the Examiner, in his remarks, has repeatedly equated "a line of instructions" (that is fetched into the instruction cache) with "a trace being evicted from the trace cache", and more specifically with the trace being evicted from the trace cache. Applicants assert that if in Applicants' claimed invention the trace cache being evicted from the trace cache and the line of instructions fetched into the instruction cache were the same thing, Applicants would be required to recite this relationship in the claims and to provide support in the specification. The fact that Applicants' claim 1 recites these two different terms in this limitation, and in the corresponding descriptions in the specification, serves to distinguish the two elements of the claim from each other. Similarly, the Examiner's own cited reference clearly distinguishes between instruction sequences stored in an instruction cache, and traces built from these instruction sequences that are stored in trace caches, as discussed above regarding paragraphs [0019] – [0021]. **Therefore, Applicants assert that the Examiner's interpretation that these two elements (the line of instructions fetched into the instruction cache and the evicted trace) are the same entity (i.e., the evicted trace) is clearly not a reasonable one.**

In the Office Action mailed September 7, 2006, the Examiner submits, "Mendelson also discloses the prefetch unit configured to fetch instruction code from a system memory for storage within the instruction cache (Paragraph 39, lines 3-6)." The

Examiner further asserts that the FTC and MTC are considered to be system memory and notes a definition from the American Heritage Dictionary, 4th addition. However, the Examiner's definition is not a definition of "system memory" as he suggests. The definition provided by the Examiner is reproduced below:

> Memory: Also called computer memory, storage.
>
> a) the capacity of a computer to store information subject to recall.
>
> b) The components of the computer in which such information is stored.

The Examiner submits, "By this definition, the FTC and MTC are clearly system memory." This is incorrect. The Examiner has not quoted a definition of "system memory" but of just "memory." The term "system memory" is well known to those of ordinary skill in the art and is used in specific contexts within the art. The Examiner's own cited reference both illustrates and describes system memory as being distinct from any of the caches contained in the processor of Mendelson (including instruction cache 140 and all of the trace caches of trace cache subsystem 120). In fact, the term "system memory", as used in Mendelson, exclusively refers to memory that is completely outside the processor. See, for example, FIG. 1A and its description in paragraphs [0013], [0015], and [0016], reproduced below, which contain the only references to the term "system memory" in Mendelson.

> [0013] FIG. 1A is a diagram illustrating a computer system 100 in which one embodiment of the present invention can be practiced. The computer system 100 includes a processor 105, a host bus 111, a host bridge chipset 121, a system memory 132, a peripheral component interconnect (PCI) bus 151, PCI slots 161, a PCI-to-industry standard architecture (ISA) bridge 172, mass storage device 173, Input/Output (I/O) ports 171, an ISA bus 182, and ISA slots 181.
>
> [0015] The host bridge chipset 121 includes a number of interface circuits to allow the host processor 105 accesses to the system memory 132 and the PCI bus 151. The system memory 132 represents one or more mechanisms for storing information. The system memory 132 may contain program 131, and other programs and data 138. The PCI slots 161 provide interfaces to PCI devices. Examples of PCI devices include the network interface and the media interface.
>
> [0016] The PCI-to-ISA bridge 172 provides access to the ISA bus 182, mass storage devices 173, and input/output (I/O) ports 171. The I/O ports 171 provides interface to the I/O devices. The mass storage device 173 stores archive

information such as code, programs, files, data, applications and operating systems. The mass storage device 173 provides a mechanism to read machine-readable media. The ISA bus 182 has a number of ISA slots 181 to interface to ISA devices. In a system where multiple processors are utilized, it is typical for multiple processors to share the main system memory 132 and/or mass storage device 173.

In light of the above, the Examiner's interpretation that FTC 320 and MTC 330 of the Mendelson reference may be considered "system memory" is clearly not a reasonable one. Therefore, the Examiner's conclusion, "Consequently, Mendelson discloses a prefetch unit (Cache Manager) configured to fetch instruction code (trace information) from a system memory (FTC or MTC) for storage within the instruction cache (L2 cache)" is clearly unsupported by the teachings of the Examiner's own reference.

Applicants remind the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical** invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Mendelson fails to disclose *the prefetch unit is configured to fetch instruction code from a system memory for storage within the instruction cache, and wherein the prefetch unit is further configured to fetch a line of instructions into the instruction cache in response to a trace being evicted from the trace cache*. Therefore, Mendelson cannot be said to anticipate claim 1.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested.

Claim 8 includes limitations similar to claim 1, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 7, contrary to the Examiner's assertion, Mendelson fails to teach or suggest *the prefetch unit is configured to inhibit the fetch of a line of instructions into the instruction cache in response to the eviction of certain traces from the trace cache if the evicted trace is predicted unlikely to re-execute.* In the previous Office Action, the Examiner asserted that Mendelson's use counter is a method of indicating if a trace is likely to be re-used (paragraph [0040], lines 5-7). While this is true, the response in Mendelson to predicting that a trace is unlikely to re-execute is not to *inhibit the fetch of a line of instructions into the instruction cache*, as in claim 7. As discussed above, the system of Mendelson is not configured to fetch a line of instructions into the instruction cache in response to a trace eviction. It is also not configured to inhibit such a fetch in response to eviction of certain traces. Instead, in Mendelson, the response to predicting that a trace is unlikely to re-execute is to discard the trace (see, e.g., FIG. 5).

In the Response to Arguments section of the Office Action mailed on September 7, 2006, the Examiner submits, "The arguments made with regard to claims 2, 3, 4, 7, and 15 all seem to ignore the analogous nature of the instruction cache (as claimed) and the L2 cache (as disclosed by Mendelson). Examiner asserts that if that fact is considered, the remaining limitations fall into place and are adequately described within the original rejection." Applicants assert, however, that, as discussed above, the Examiner's interpretation that the instruction cache of Applicants' claims is analogous to the L2 trace cache of Mendelson is not a reasonable one and is unsupportable in light of his own cited reference. Applicants further assert that the Examiner's interpretation that the prefetch unit of Applicants' claims is analogous to the Cache Manager of Mendelson, that the line of instructions fetched into the instruction cache and the evicted trace cache are the same element, and that the trace caches of Mendelson may be considered the system memory of Applicants' claims are equally unsupportable, given the description of these elements in the Examiner's own cited reference.

For at least the reasons above, the rejection of claim 7 is not supported by the cited art and removal thereof is respectfully requested.

Claims 14 and 18 include limitations similar to claim 7, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 15, contrary to the Examiner's assertion, Mendelson fails to teach or suggest *evicting a trace from a trace cache* and *fetching a line of instructions into an instruction cache from a system memory in response to said evicting*. The Examiner again cited paragraph [0039], lines 3-6 as teaching these limitations. However, as discussed above, Mendelson does not disclose fetching a line of instructions, or fetching them into an instruction cache from a system memory, as recited in claim 15. Instead, Mendelson describes transferring an evicted trace entry itself from one trace cache to another trace cache. Applicants again assert that the Examiner's analogies regarding the instruction cache, the prefetch unit, system memory, and the line of instructions fetched into the instruction cache, discussed above, are clearly unsupported by his own cited reference. Therefore, Mendelson cannot be said to anticipate claim 15.

For at least the reasons above, the rejection of claim 15 is not supported by the cited art and removal thereof is respectfully requested.

Claim 23 includes limitations similar to claim 15, and so the arguments presented above apply with equal force to this claim, as well.

Claims 13 and 17 were rejected under 35 U.S.C. § 102(b) as being anticipated by Mendelson. However, the Examiner's only remarks regarding these rejections (in the Office Action mailed September 7, 2006) state that these claims were rejected under the same grounds as claim 6. However, claim 6 was rejected under 35 U.S.C. § 103(a), not 35 U.S.C. § 102(b). The Examiner failed to present any other remarks supporting the rejection of claims 13 and 17 under 35 U.S.C. § 102(b) in the present Office Action. Applicants note MPEP 707.07(d), which requires that, in an Examiner's Action, the ground of rejection, should be "fully and clearly stated". Since the Examiner has not fully and clearly stated any line of reasoning to support the rejection of claims 13 and 17 under 35 U.S.C. § 102(b), Applicants assert that the rejection of these claims is improper.

Section 103(a) Rejection:

The Examiner rejected claims 2-6, 9-12, 16 and 19-22 under 35 U.S.C. § 103(a) as being unpatentable over Mendelson. Applicants traverse this rejection for at least the following reasons.

Regarding claim 2, contrary to the Examiner's assertion, Mendelson does not teach or suggest *the prefetch unit is configured to fetch a line into the instruction cache comprising instructions that correspond to operations that precede a branch in the evicted trace.* In the previous Office Action, the Examiner took Official Notice that traces may consist of multiple branches and asserts that if a trace has two or more branch instructions in it, it is guaranteed to have at least one instruction which precedes a branch instruction. Applicants asserted, however, that claim 2 recites fetching a line into the instruction cache (Mendelson's cache memory 140) comprising instructions that correspond to operations that precede a branch in the evicted trace, which the system of Mendelson does not do. Instead, Mendelson discloses transferring an evicted trace from one trace cache to another trace cache. Therefore, the Examiner's remarks do not apply to claim 2, regardless of the contents of the evicted trace. In addition, as discussed above, the Examiner's analogies regarding the instruction cache, the prefetch unit, and the line of instructions fetched into the instruction cache have been shown to be unsupportable in his own cited reference.

For at least the reasons above, the rejection of claim 2 is not supported by the cited art and removal thereof is respectfully requested.

Claims 9 and 19 include limitations similar to claim 2, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 3, contrary to the Examiner's assertion, Mendelson does not teach or suggest *the prefetch unit is configured to fetch a line into the instruction cache comprising instructions that correspond to operations that follow a branch in the evicted trace.* In the previous Office Action, the Examiner took Official Notice that traces may consist of multiple branches and asserts that if a trace has two or more branch instructions in it, it is guaranteed to have at least one instruction which follows a branch instruction. Applicants argued, however, that claim 3 recites fetching a line into the instruction cache (Mendelson's cache memory 140) comprising instructions that correspond to operations that follow a branch in the evicted trace, which the system of Mendelson does not do. Instead, Mendelson discloses transferring an evicted trace from one trace cache to another trace cache. Therefore, the Examiner's remarks do not apply to claim 3, regardless of the contents of the evicted trace. In addition, as discussed above, the Examiner's analogies regarding the instruction cache, the prefetch unit, and the line of instructions fetched into the instruction cache have been shown to be unsupportable in his own cited reference.

For at least the reasons above, the rejection of claim 3 is not supported by the cited art and removal thereof is respectfully requested.

Claims 10 and 20 include limitations similar to claim 3, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 4, in the previous Office Action, the Examiner admitted that Mendelson fails to disclose *wherein the prefetch unit is configured to prefetch a plurality of lines of instructions into the instruction cache in response to the trace being evicted from the trace cache.* The Examiner took Office notice that traces commonly consist of multiple lines when stored in a cache. Applicants argued, however, that claim 4 recites *prefetching a plurality of lines of instructions into the instruction cache* (Mendelson's cache memory 140) *in response to the trace being evicted from the trace cache*, which the system of Mendelson does not do. Instead, Mendelson discloses transferring an evicted trace from one trace cache to another trace cache. Therefore, the Examiner's

remarks do not apply to claim 4, regardless of the contents of the evicted trace or the number of lines in which it is stored. In addition, as discussed above, the Examiner's analogies regarding the instruction cache, the prefetch unit, and the line of instructions fetched into the instruction cache have been shown to be unsupportable in his own cited reference.

For at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested.

Claims 11 and 21 include limitations similar to claim 4, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 16, contrary to the Examiner's assertion, Mendelson fails to teach or suggest *checking the instruction cache for lines of instructions comprising the instructions corresponding to the evicted trace.* In the previous Office Action, the Examiner rejected this claim under 35 U.S.C. § 102(b), and asserted that the processor must inherently check if a trace is already stored in the L2 cache before storing it and that storing the same trace cache more than once in the L2 cache is a waste of resources. Applicants respectfully disagreed. First, Applicants' claim does not recite checking a trace cache (such as Mendelson's L2) for lines of instructions comprising the instructions corresponding to the evicted trace, but instead recites checking the instruction cache (Mendelson's cache memory 140).

Applicants further argued that there is nothing inherent about checking any memory (including an instruction cache or a trace cache) to see if an item is already stored there before storing it, that the Examiner provided no such evidence, and that, in fact, his assertion is incorrect. As would be clear to one of ordinary skill in the art at the time the invention was made, in various embodiments of a computer system or its memory systems, there may be many valid reasons for storing a duplicate copy of something in a memory, such as to provide redundancy, or to optimize the system for speed of storing data (such as by storing data without such checking) rather than for

memory utilization (not "wasting resources"), just to name two. Since Mendelson does not teach or suggest *checking the instruction cache for lines of instructions comprising the instructions corresponding to the evicted trace*, and it was shown that this is not inherent in the implementation of a trace cache or instruction cache, Applicants argued that Mendelson cannot be said to anticipate claim 16.

In the Response to Arguments section of the Office Action mailed September 7, 2006, the Examiner states, "Examiner agrees. This aspect is not inherent. The rejection has been changed to an obvious rejection under 35 USC 103."

In the Office Action mailed September 7, 2006, the Examiner states, in rejecting claim 16 under 35 U.S.C. § 103(a), "Mendelson discloses the method of claim 15, further comprising checking the instruction cache for lines of instructions comprising the instructions corresponding to the evicted trace. See claim 6." Applicants note that the scope of claim 16 is not the same as claim 6 and recites limitations not found in claim 6. Therefore, the Examiner has failed to provide a fully and clearly stated ground of rejection for claim 16. In addition, Applicants assert that Mendelson does not teach or suggest the limitations of claim 6. Both of these arguments are discussed below.

Claim 16 recites, "The method of claim 15, further comprising checking the instruction cache for lines of instructions comprising the instructions corresponding to the evicted trace." Claim 6 recites, "The microprocessor of claim 1, wherein the prefetch unit is configured to inhibit the fetch of a line of instructions into the instruction cache in response to the eviction of certain traces from the trace cache if the line of instructions is already stored in the instruction cache." Thus, claim 6 is directed toward inhibiting fetching of a line of instructions if it (the line of instructions) is already in the instruction cache. Claim 16 is not directed toward checking the instruction cache for a line of instructions that might otherwise be fetched, but is directed toward checking the instruction cache for lines of instructions comprising instructions corresponding to **an evicted trace**. As discussed above regarding claim 1, in Mendelson, a line of instructions fetched into an instruction cache is not the same as a trace stored in (or evicted from) a trace cache. Traces are built from (but are not the same as) instruction sequences or lines

of instructions. Therefore, claim 16 is directed toward checking the instruction cache for lines of instructions comprising instructions (as stored in the instruction cache) corresponding to the evicted trace (which would have been built from, but is not the same as, the instruction sequence from which the instructions were originally fetched into the instruction cache.) Applicants assert that there is nothing in Mendelson that teaches checking an instruction cache (i.e., instruction cache 140) for lines of instructions comprising instructions corresponding to an evicted trace (which is not the same as the line of instructions itself.) As discussed above, the Examiner's analogies regarding the instruction cache, the prefetch unit, and the line of instructions fetched into the instruction cache have been shown to be unsupportable in his own cited reference.

For at least the reasons above, the rejection of claim 16 is not supported by the cited art and removal thereof is respectfully requested.

In regard to the rejections under both § 102(e) and § 103(a), Applicants assert that numerous other ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Applicants hereby petition for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-91600/RCK.

Also enclosed herewith are the following items:

☐ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Other:

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: December 7, 2006